http://www.cs.unh.edu/~ruml/cs758

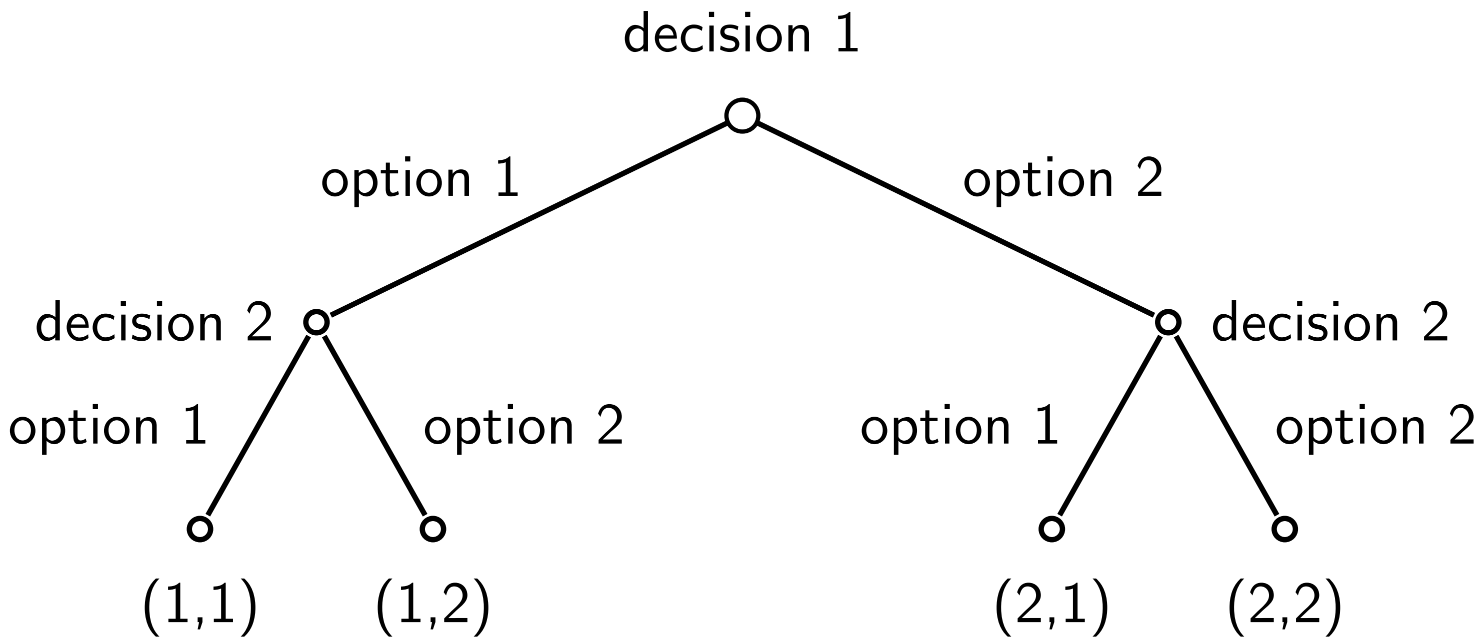
Backtracking

Hardness

NPC: SAT, vertex cover, clique, subset sum, ...

greedy: local choice is optimal
DP: poly number of options to track
search: exponential number of options, often combinations

Combinatorial Optimization



A tree representation of alternatives in a small combinatorial problem.

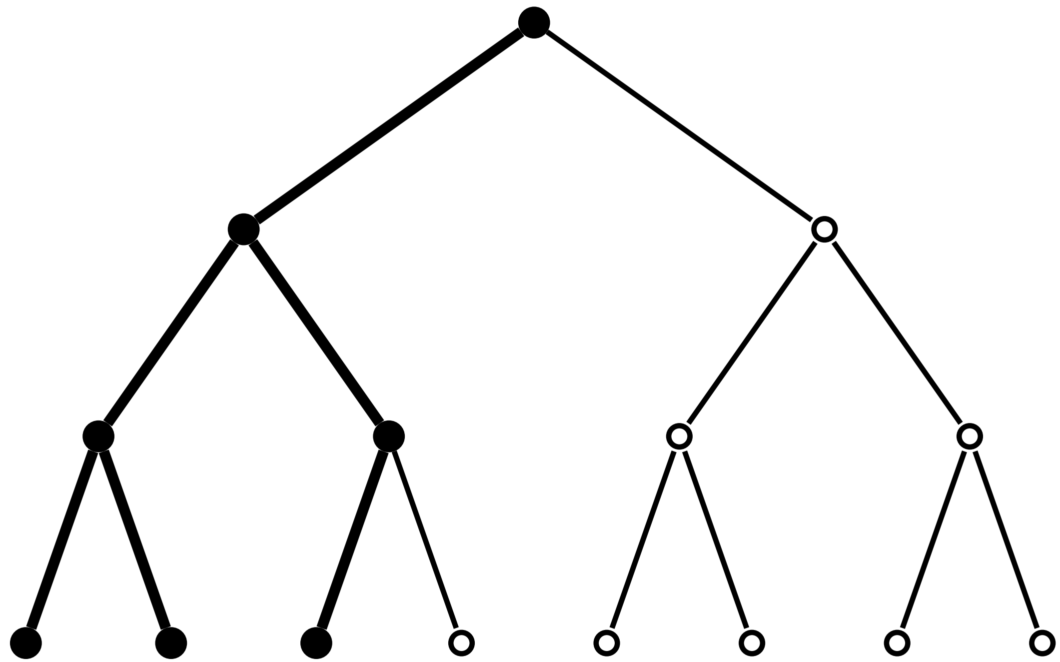
Backtracking

depth-first search
child ordering
lower bounds
branch-and-bound
duplicate detection: transposition table

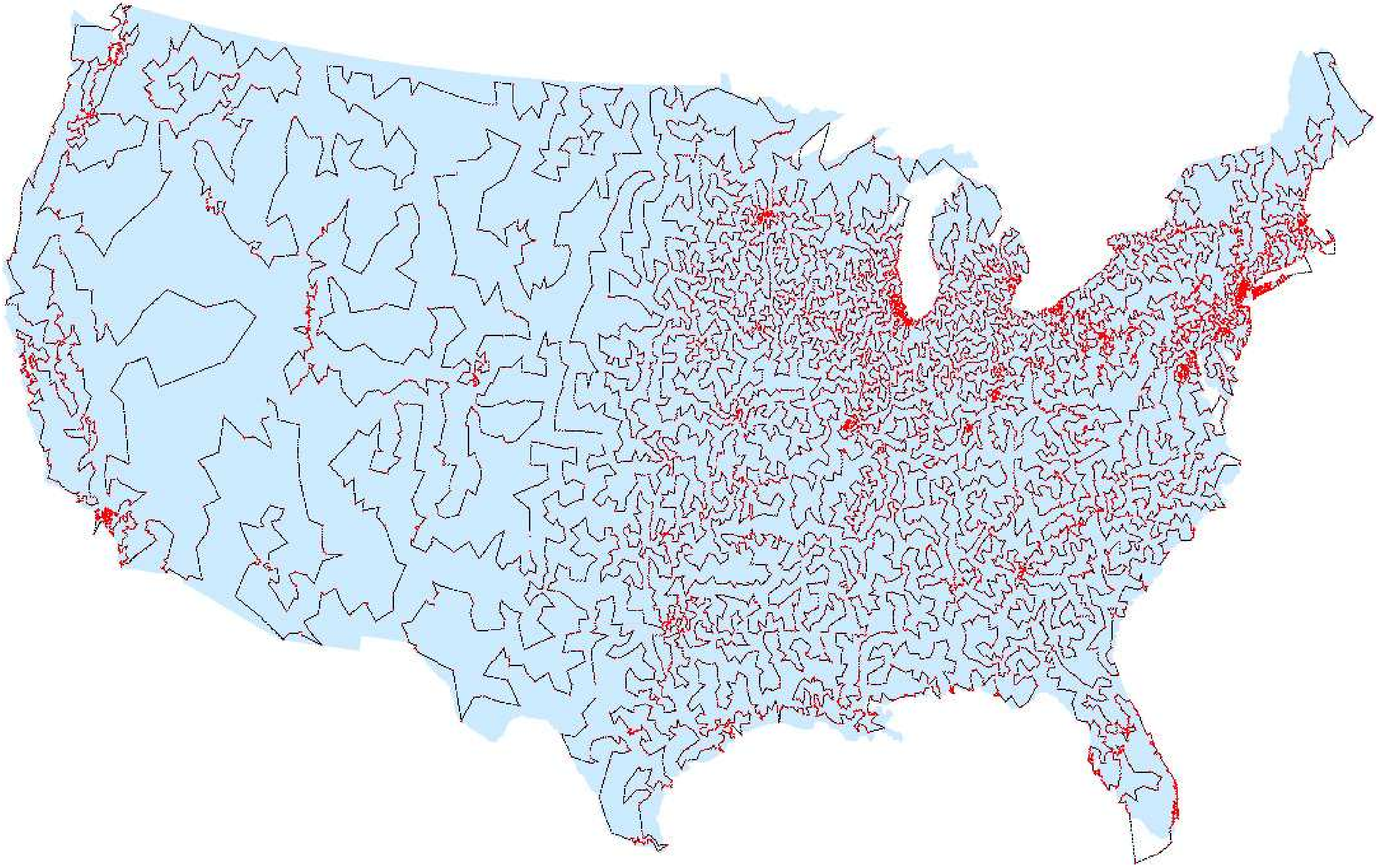
Depth-first Search


```
DFS (node)
    If is-leaf(node)
        Visit(node)
    else
        For i from 0 to num-children
            DFS(child(node, i))
```

Depth-first Search Order



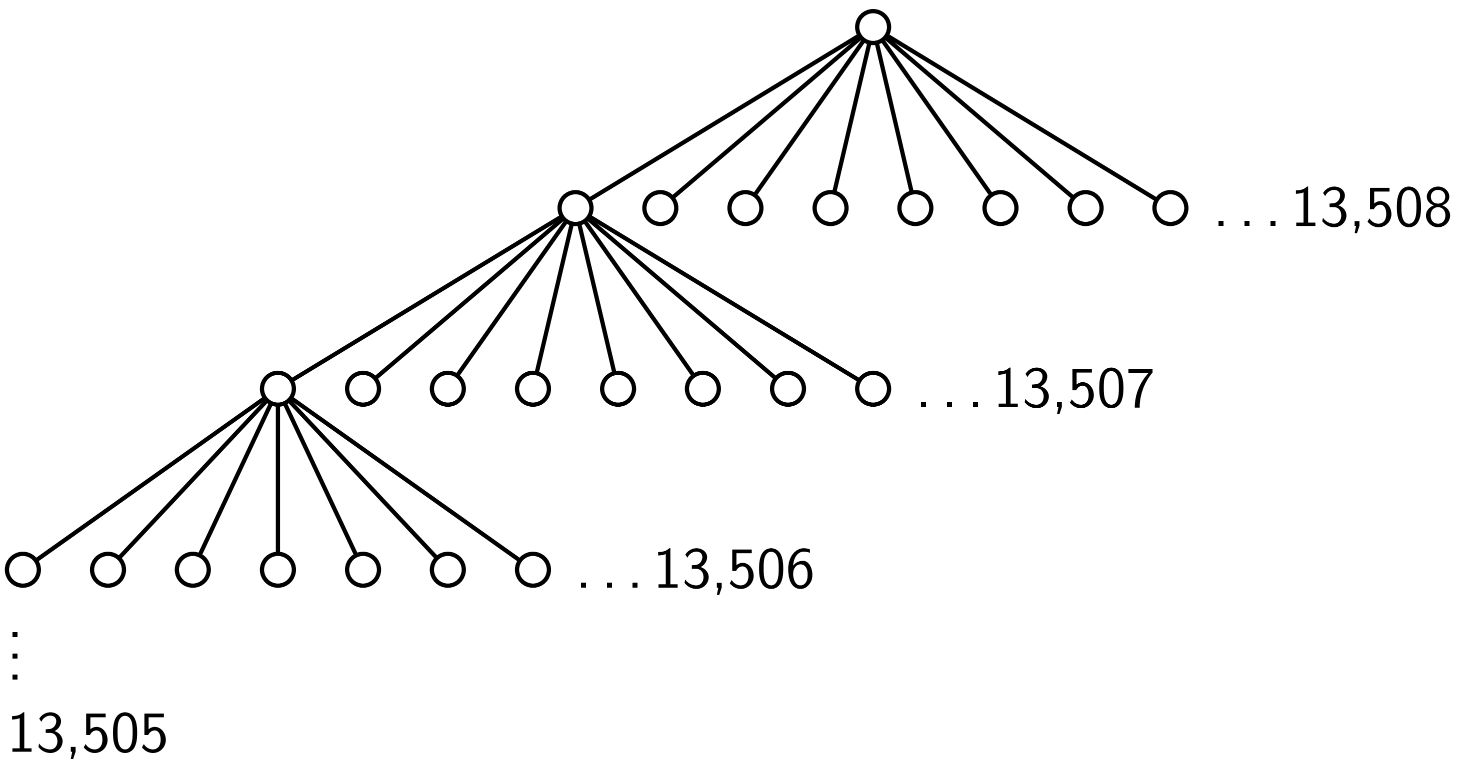
Problems Are Hard



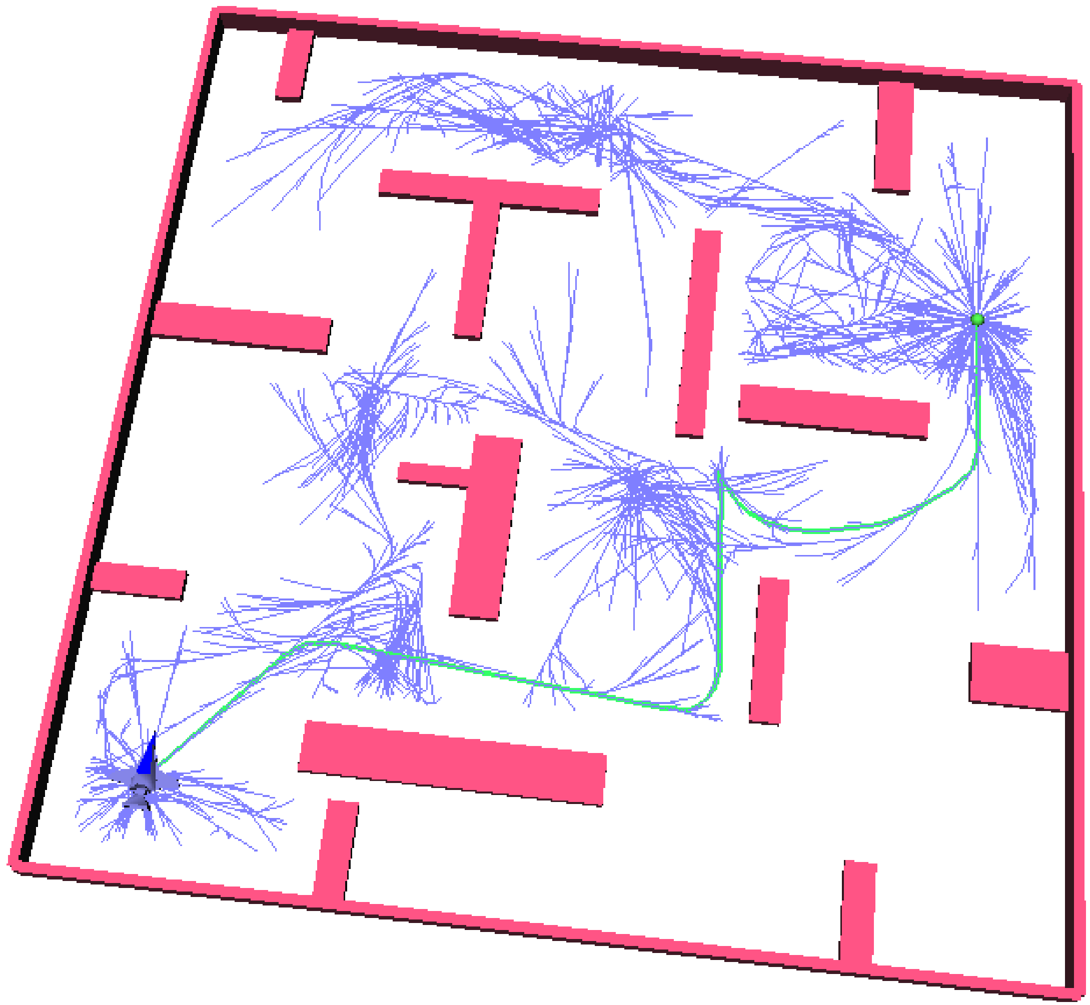
13,509 US cities (W. Cook)

largest TSP known solved is 85,900 cities

Problems Are Hard

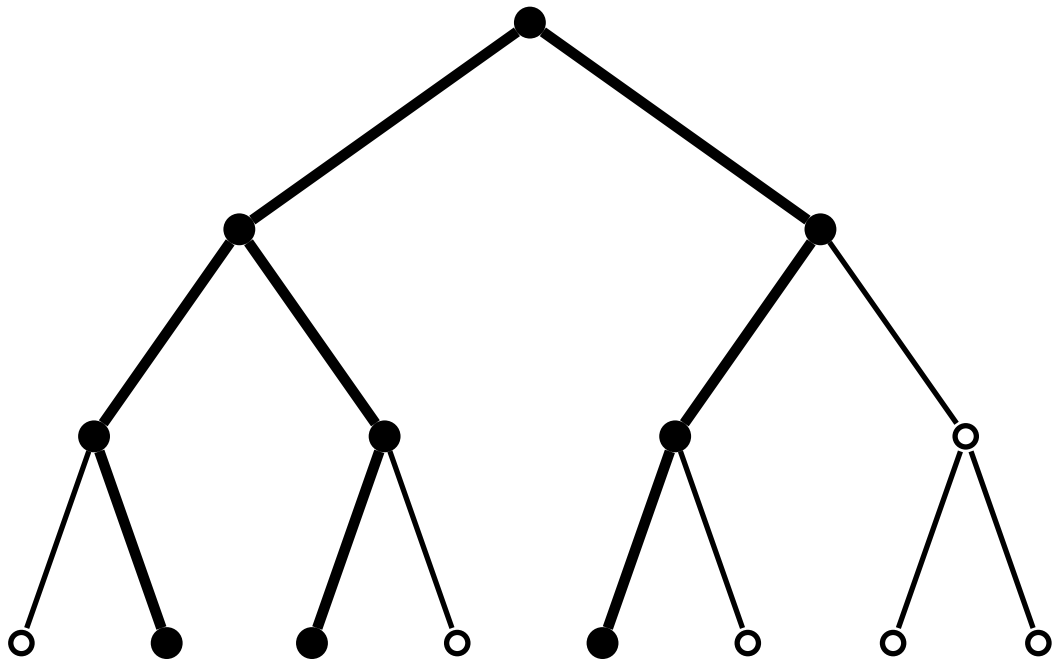


Problems Are Hard



(S. LaValle)

Discrepancy Search Order



The second pass of ILDS visits all leaves with one discrepancy in their path from the root.

Improved Discrepancy Search

ILDS (*node*, *allowance*, *remaining*)

If is-leaf(*node*)
    Visit(*node*)
else
    If $allowance > 0$
        ILDS(child(*node*, 1), $allowance - 1$, $remaining - 1$)
    If $remaining > allowance$
        ILDS(child(*node*, 0), $allowance$, $remaining - 1$)

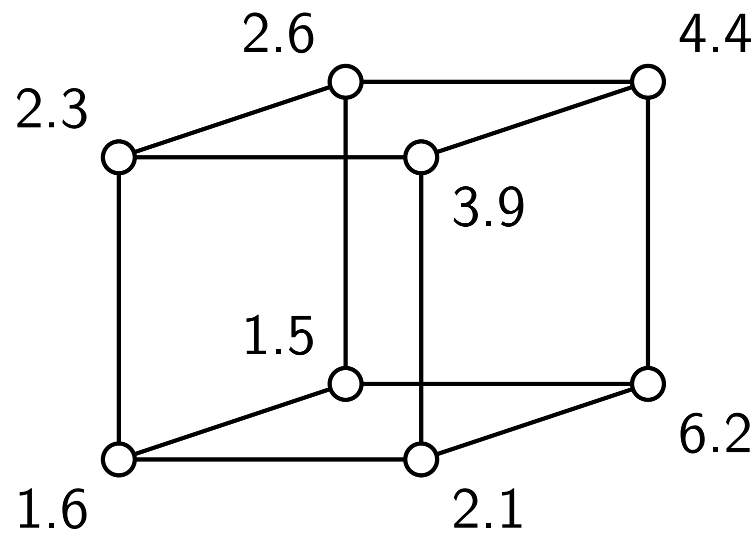
start with ILDS(root, *iteration*, *max-depth*)

Break

- asst 14
- recitation: last year's final
- final exam: Wed Dec 11, 3:30-5:30pm, Kingsbury N121
- 10 min for finishing SEL surveys on Wed: bring device!

Local Search

Local Search



A graph representing an improvement-based search.

Local Search

hill climbing
simulated annealing
large neighborhood search
genetic algorithms
particle swarm optimization

Max Cut

maximize weight of edges crossing the cut $w(A, B)$

decision version is NP-complete

simple local search:

move vertex u from A to B iff

$$\sum_{v \neq u \in A} w_{uv} > \sum_{v \in B} w_{uv}$$

it's possible to bound suboptimality of local minima under this neighborhood!

Suboptimality of Local Search

for any u in A,

$$\sum_{v \neq u \in A} w_{uv} \leq \sum_{v \in B} w_{uv}$$

summing over all u in A,

$$2 \sum_{(u,v) \in A} w_{uv} \leq \sum_{u \in A, v \in B} w_{uv} = w(A, B)$$

same from perspective of B:

$$2 \sum_{(u,v) \in B} w_{uv} \leq \sum_{u \in A, v \in B} w_{uv} = w(A, B)$$

add:

$$2 \sum_{(u,v) \in A} w_{uv} + 2 \sum_{(u,v) \in B} w_{uv} \leq 2w(A, B)$$

Suboptimality of Local Search

divide by 2:

$$\sum_{(u,v)\in A} w_{uv} + \sum_{(u,v)\in B} w_{uv} \leq w(A,B)$$

eg, more weight crossing than within partitions

let W be sum of all weight in graph.
add crossing weight to both sides:

$$W \leq 2w(A,B)$$

$$W/2 \leq w(A,B)$$

note optimal is at most W

EOLQs

For example:

- What's still confusing?
- What question didn't you get to ask today?
- What would you like to hear more about?

Please write down your most pressing question about algorithms and put it in the box on your way out.
Thanks!